GitLab Complete Usage Guide for Software Companies

1. Introduction

GitLab is a complete DevOps platform, delivered as a single application. It enables software teams to plan, build, secure, and deploy software collaboratively and efficiently.

2. Project Setup and Repository Management

- Create a Project: Navigate to Projects > New Project, choose blank/template/import, set visibility, and initialize with README.
- Branch Strategy: Use main for production, develop for integration, and feature branches for tasks.
- Merge Requests: Submit MRs for code reviews before merging to main.

3. User Management and Roles

- Adding Users: Go to Settings > Members to invite users.
- Roles:
  - Guest: Read-only access.
  - Reporter: View and pull code.
  - Developer: Commit and push code.
  - Maintainer: Manage settings, pipelines.
  - Owner: Full control (group level).
- Groups: Use groups for managing access across multiple projects.

4. Issue Tracking and Agile Tools

- Issues: Used for task tracking.
- Epics: Group related issues.
- Boards: Visual Kanban boards for workflows.
- Milestones & Burndown Charts: Sprint planning and tracking.

5. CI/CD Pipelines

- Basic Pipeline Structure (.gitlab-ci.yml):

```
stages:
```

```
    - build
    - test
    - deploy

  build:
    stage: build
    script: echo "Building..."

  test:
    stage: test
    script: make test

  deploy:
    stage: deploy
    script: echo "Deploying..."
```

- Runners: Shared or self-hosted to execute pipelines.
- Advanced: Use needs:, artifacts:, scheduled pipelines, manual approvals.

6. Code Review and Collaboration

- Inline discussions in MRs.
- Required approvals and protected branches.
- Review Apps for testing environments.

7. Documentation

- Wikis: Each project has a version-controlled wiki.
- GitLab Pages: Host static sites for docs.

8. Package and Container Registry

- Package Registry: Manage npm, Maven, NuGet, etc.

- Container Registry: Store Docker images linked to projects.

9. Security and Compliance

- Static & Dynamic Scanning (SAST/DAST)
- Dependency and Container Scanning
- License Compliance Checks
- Shift-left security integrated into the pipeline.

10. Deployment & Environment Management

- Environment Setup: Define staging, production, etc.
- Auto DevOps: Automatic CI/CD pipeline with Kubernetes.
- GitOps: Declarative infra with Flux/ArgoCD integration.

11. Integrations and Automation

- Webhooks & APIs: Trigger actions based on events.
- Third-party Integrations: Slack, Jira, Sentry, Trello.

12. Analytics and Dashboards

- Value Stream Analytics
- Issue Analytics and Cycle Time
- CI/CD Job Duration Metrics

13. Permissions and Governance

- Protected Branches & Environments
- Central Policy Management
- Audit Logs and Compliance Frameworks

14. Community and Tips

- Leverage GitLab forums, Reddit, and YouTube for community tips.

- Try GitLab CLI tools and integrations for productivity.

15. References

- GitLab Documentation: https://docs.gitlab.com
- CI/CD Examples: https://docs.gitlab.com/ee/ci/examples/
- DevSecOps Resources: https://about.gitlab.com/topics/devsecops/